

LAPORAN TUGAS PEMROGRAMAN BERBASIS FRAMEWORK

Modul 5 React Router



Disusun Oleh :

Nama : Hunayn Risatayn
NIM : 1841720148
Kelas : TI-3E

Program Studi D-IV Teknik Informatika
Jurusan Teknologi Informasi
Politeknik Negeri Malang
Maret 2021

Praktikum – Bagian 1: Basic Router

Langkah	Keterangan
1	<p>Setelah kita sukses melakukan instalasi react (npm install), kita perlu menambahkan library react router dengan menambahkan npm install react-router-dom.</p> <pre> PS D:\data\Users\User\Documents\College\Framework Prog> cd "D:\data\Users\User\Documents\College\Framework Prog\framework_prog\5thWeek" npx: installed 67 in 14.012s Creating a new React app in D:\data\Users\User\Documents\College\Framework Prog\framework_prog\5thWeek\react-router-hunayn. Installing packages. This might take a couple of minutes. Installing react, react-dom, and react-scripts with cra-template... [.....] - loadDep:picomatch: 3.1.1 resolveWithNewModule sane@4.1.0 checking installable status [> core-js@2.6.12 postinstall D:\data\Users\User\Documents\College\Framework Prog\framework_prog\5thWeek\react-router-hunayn\node_modules\babel-run time\node_modules\core-js > node -e "try{require('./postinstall')}catch(e){}" Starts the test runner. npm run eject Removes this tool and copies build dependencies, configuration files and scripts into the app directory. If you do this, you can't go back! We suggest that you begin by typing: cd react-router-hunayn npm start Happy hacking! PS D:\data\Users\User\Documents\College\Framework Prog\framework_prog\5thWeek> </pre> <pre> PS D:\data\Users\User\Documents\College\Framework Prog\framework_prog\5thWeek> cd react-router-hunayn PS D:\data\Users\User\Documents\College\Framework Prog\framework_prog\5thWeek\react-router-hunayn> npm install react-router-dom npm WARN tsutils@3.21.0 requires a peer of typescript@>=2.8.0 >= 3.2.0-dev >= 3.3.0-dev >= 3.4.0-dev >= 3.5.0-dev >= 3.6.0-dev > = 3.6.0-beta >= 3.7.0-dev >= 3.7.0-beta but none is installed. You must install peer dependencies yourself. npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents): npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32" ,"arch":"x64"}) npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\watchpack-chokidar2\node_modules\fsevents): npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32 ","arch":"x64"}) npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-dev-server\node_modules\fsevents): npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32 ","arch":"x64"}) + react-router-dom@5.2.0 added 11 packages from 6 contributors and audited 1960 packages in 18.313s 131 packages are looking for funding run `npm fund` for details found 0 vulnerabilities PS D:\data\Users\User\Documents\College\Framework Prog\framework_prog\5thWeek\react-router-hunayn> </pre> <p>Kemudian buka app.js pada dan ketikkan code dibawah ini.</p> <pre> framework_prog > 5thWeek > react-router-hunayn > src > JS App.js > ... 1 import React from "react"; 2 import { BrowserRouter as Router, Switch, Route, Link } from "react-router-dom"; 3 4 // Situs ini memiliki 3 halaman, yang semuanya dirender secara dinamis di browser 5 // Meskipun halaman tidak pernah di-refresh, perhatikan bagaimana React Router 6 // membuat URL selalu terbaru saat anda menavigasi situs. 7 // Ini menjaga riwayat browser, memastikan hal-hal seperti tombol kembali 8 // dan bookmark berfungsi dengan baik. 9 10 export default function BasicExample() { 11 return (12 <Router> 13 <div> 14 15 16 <Link to="/">Home</Link> 17 18 19 <Link to="/about">About</Link> 20 21 22 <Link to="/dashboard">Dashboard</Link> 23 24 </pre>

```

24     </ul>
25     <hr />
26
27     <Switch>
28       <Route exact path="/">
29         <Home />
30       </Route>
31       <Route path="/about">
32         <About />
33       </Route>
34       <Route path="/dashboard">
35         <Dashboard />
36       </Route>
37     </Switch>
38   </div>
39 </Router>
40 );
41 }
42
43 // Anda dapat menganggap komponen ini sebagai "halaman(konten)" di situs anda.
44
45 function Home() {
46   return (
47     <div>

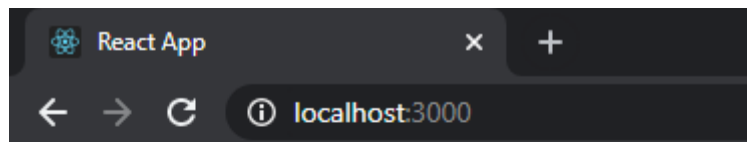
```

```

47       <div>
48         <h2>Home</h2>
49       </div>
50     );
51   }
52
53   function About() {
54     return (
55       <div>
56         <h2>About</h2>
57       </div>
58     );
59   }
60
61   function Dashboard() {
62     return (
63       <div>
64         <h2>Dashboard</h2>
65       </div>
66     );
67   }
68
69   // function App() {
70   //   return (

```

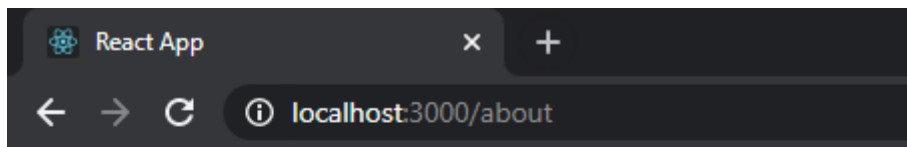
Lakukan npm start, lalu cek hasilnya akan seperti dibawah ini.



- [Home](#)
- [About](#)
- [Dashboard](#)

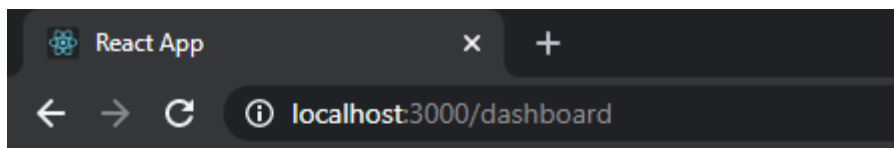
Home

Pilih setiap menu lalu amati bagaimana perubahannya.



- [Home](#)
- [About](#)
- [Dashboard](#)

About



- [Home](#)
- [About](#)
- [Dashboard](#)

Dashboard

URL Parameters

Pada percobaan kali ini kita mencoba membuat link react router dengan menggunakan params, sehingga kita hanya membuat satu template yang dapat berubah kontennya sesuai dengan apa yang kita klik.

Ketikkan seperti code dibawah ini.

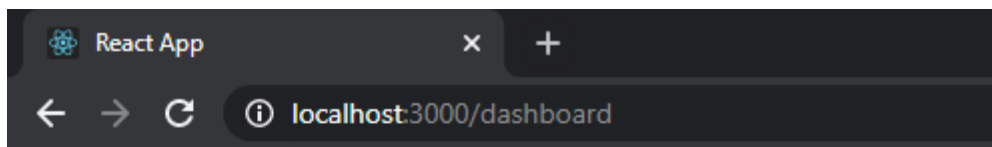
```
framework_prog > 5thWeek > react-router-hunayn > src > JS App.js > ...
1  import React from "react";
2  import {
3    BrowserRouter as Router,
4    Switch,
5    Route,
6    Link,
7    useParams,
8  } from "react-router-dom";
9
10 // Params adalah placeholder di URL yang didahului dengan titik dua,
11 // seperti param `: id` yang didefinisikan dalam route dalam contoh ini
12
13 export default function ParamsExample() {
14   return (
15     <Router>
16       <div>
17         <h2>Accounts</h2>
18         <ul>
19           <li>
20             <Link to="/netflix">Netflix</Link>
21           </li>
22           <li>
23             <Link to="/gmail">Gmail</Link>
24           </li>
25           <li>
26             <Link to="/yahoo">Yahoo</Link>
27           </li>
28           <li>
```

```

26 |         <Link to="/yahoo">Yahoo</Link>
27 |       </li>
28 |       <li>
29 |         <Link to="/amazon">Amazon</Link>
30 |       </li>
31 |     </ul>
32 |
33 |     <Switch>
34 |       <Route path="/:id" children={<Child />} />
35 |     </Switch>
36 |   </div>
37 </Router>
38 );
39 }
40
41 function Child() {
42   let { id } = useParams();
43
44   return (
45     <div>
46       <h3>ID: {id}</h3>
47     </div>
48   );
49 }
50

```

Cek hasilnya maka akan seperti dibawah ini.



Accounts

- [Netflix](#)
- [Gmail](#)
- [Yahoo](#)
- [Amazon](#)

ID: dashboard

Secara tampilan memang hasilnya hampir sama dengan praktikum sebelumnya, namun dengan metode yang berbeda.

Use Nesting Router

Jika kita ingin melakukan percabangan link di dalam react router, maka kita dapat menggunakan cara ini sebagai solusinya.

Ketikkan seperti code dibawah ini

```

framework_prog > 5thWeek > react-router-hunayn > src > JS App.js > Topic
1  import React from "react";
2  import {
3    BrowserRouter as Router,
4    Switch,
5    Route,
6    Link,
7    useParams,
8    useRouteMatch,
9  } from "react-router-dom";
10
11 // Karena route adalah komponene regular react,
12 // sehingga dapat ditampilkan di mana saja dalam penempatannya,
13 // termasuk dalam child element.
14
15 // Ini akan membantu anda untuk memecah menjadi beberapa bundle
16 // karena pemisahan kode pada aplikasi React Router sama dengan
17 // pemecahan kode pada aplikasi react lainnya
18
19 export default function NestingExample() {
20   return (
21     <Router>
22       <div>
23         <ul>
24           <li>
25             <Link to="/">Home</Link>
26           </li>
27           <li>
28             <Link to="/topics">Topics</Link>
29             <Link to="/topics">Topics</Link>
30           </li>
31         </ul>
32         <hr />
33         <Switch>
34           <Route exact path="/topics">
35             <Topics />
36           </Route>
37         </Switch>
38       </div>
39     </Router>
40   );
41 }
42
43
44 function Home() {
45   return (
46     <div>
47       <h2>Home</h2>
48     </div>
49   );
50 }
51
52 function Topics() {
53   // `path` untuk membuat jalur <Route> yang terhadap rute induk,
54   // sedangkan `url` untuk membuat link.
55   let { path, url } = useRouteMatch();

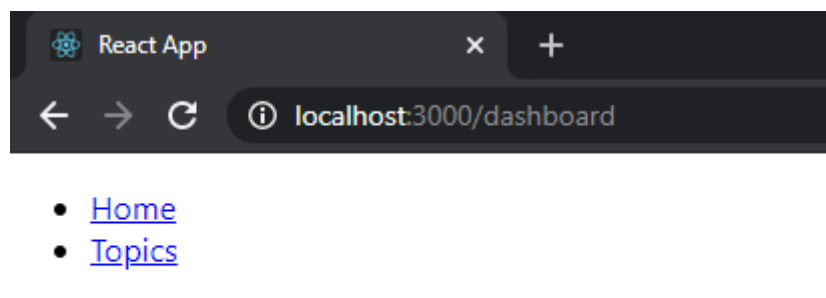
```

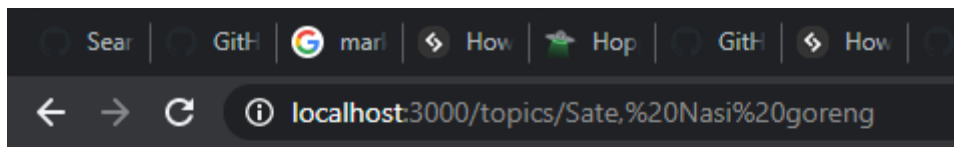
```

55   let { path, url } = useRouteMatch();
56   return (
57     <div>
58       <h2>Topics</h2>
59       <ul>
60         <li>
61           <Link to={`${url}/Sate, Nasi goreng`}>Kuliner</Link>
62         </li>
63         <li>
64           <Link to={`${url}/Wisata alam, Museum`}>Traveling</Link>
65         </li>
66         <li>
67           <Link to={`${url}/Ibis, JW Marriott`}>Review Hotel</Link>
68         </li>
69       </ul>
70
71       <Switch>
72         <Route exact path={path}>
73           <h3>Please select a topic</h3>
74         </Route>
75         <Route path={`${path}/:topicId`} >
76           <Topic />
77         </Route>
78       </Switch>
79     </div>
80   );
81 }
82
83
84
85
86
87
88
89
90
91
92

```

Cek hasilnya dan coba klik setiap link nya.





- [Home](#)
- [Topics](#)

Topics

- [Kuliner](#)
- [Traveling](#)
- [Review Hotel](#)

Sate, Nasi goreng

Bisa kita lihat bahwa link induk yang diatas akan menentukan hasil yang keluar dibawahnya, dan link yang dibawah dapat memunculkan hasil yg berbeda pula

Use Redirects (Auth)

Cara ini dapat anda gunakan untuk masuk ke halaman yang membutuhkan authentication. Setelah masuk dan melakukan klik ke menu lain termasuk masuk ke halaman yang memerlukan authentication, posisi user tidak perlu login ulang (karena posisisudah logged) sampai user malakukan sign out.

Ketikkan seperti code dibawah in.

```
framework_prog > 5thWeek > react-router-hunayn > src > JS App.js > PrivateRoute
1  import React from "react";
2  import {
3    BrowserRouter as Router,
4    Switch,
5    Route,
6    Link,
7    Redirect,
8    useHistory,
9    useLocation,
10 } from "react-router-dom";
11
12 // Pada aplikasi ini memiliki 3 halaman: public page, private page, dan halaman login.
13 // Untuk masuk ke private page, Anda harus login terlebih dahulu.
14
15 // Pertama, klik public page. kemudian, kunjungi private page.
16 // karena anda belum login, jadi anda diarahkan ke halaman login.
17 // Setelah login, Anda akan diarahkan kembali ke private page.
18
19 // Perhatikan perubahan setiap URL. Jika anda mengklik tombol kembali,
20 // apakah anda kembali ke halaman login? Tidak, karena anda sudah login.
21 // Cobalah, maka anda akan kembali ke halaman yang anda kunjungi sebelum login, yaitu public page
22
23 export default function AuthExample() {
24   return (
25     <Router>
26       <div>
27         <AuthButton />
28
```



```

28
29
30     <li>
31       <Link to="/public">Public Page</Link>
32     </li>
33     <li>
34       <Link to="/private">Private Page</Link>
35     </li>
36   </ul>
37
38   <Switch>
39     <Route path="/public">
40       <PublicPage />
41     </Route>
42     <Route path="/login">
43       <LoginPage />
44     </Route>
45     <PrivateRoute path="/private">
46       <ProtectedPage />
47     </PrivateRoute>
48   </Switch>
49 </div>
50 </Router>
51 );
52 }
53
54 const fakeAuth = {
55   isAuthenticated: false,
56   authenticated(cb) {
57     fakeAuth.isAuthenticated = true;
58     setTimeout(cb, 100); // fake async
59   },
60   signout(cb) {
61     fakeAuth.isAuthenticated = false;
62     setTimeout(cb, 100);
63   },
64 };
65
66 function AuthButton() {
67   let history = useHistory();
68
69   return fakeAuth.isAuthenticated ? (
70     <p>
71       Welcome!{" "}
72       <button
73         onClick={() => {
74           fakeAuth.signout(() => history.push("/"));
75         }}
76       >
77         Sign Out
78       </button>
79     </p>
80   ) : (
81     <p>You are not logged in</p>

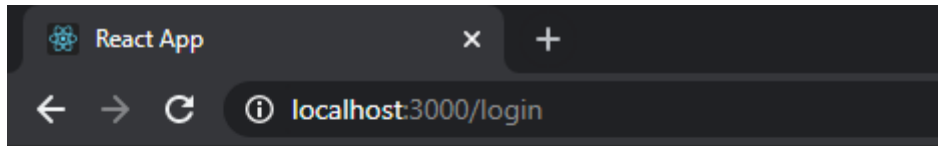
```

```

81 |     <p>You are not logged in</p>
82 |   );
83 | }
84 |
85 | // Pembungkus untuk <Route> yang mengarahkan ke login
86 | // tampilan jika anda belum terkonfirmasi
87 |
88 | function PrivateRoute({ children, ...rest }) {
89 |   return (
90 |     <Route
91 |       {...rest}
92 |       render={({ location }) =>
93 |         fakeAuth.isAuthenticated ? (
94 |           children
95 |         ) : (
96 |           <Redirect
97 |             to={{
98 |               pathname: "/login",
99 |               state: { from: location },
100 |             }}
101 |           />
102 |         )
103 |       )
104 |     />
105 |   );
106 | }
107 |
108 | function PublicPage() {
109 |   return <h3>Public</h3>;
110 | }
111 |
112 | function ProtectedPage() {
113 |   return <h3>Private</h3>;
114 | }
115 |
116 | function LoginPage() {
117 |   let history = useHistory();
118 |   let location = useLocation();
119 |
120 |   let { from } = location.state || { from: { pathname: "/" } };
121 |   let login = () => {
122 |     fakeAuth.authenticate(() => {
123 |       history.replace(from);
124 |     });
125 |   };
126 |
127 |   return (
128 |     <div>
129 |       <p>You must log in to view the page at {from.pathname}</p>
130 |       <button onClick={login}>Log in</button>
131 |     </div>
132 |   );
133 | }
134 |

```

Cek hasilnya, ketika klik private page maka diharuskan untuk login terlebih dahulu, ketika sudah masuk dan klik private page kembali kita tidak perlu login kembali karena kita berada dalam posisi logged.

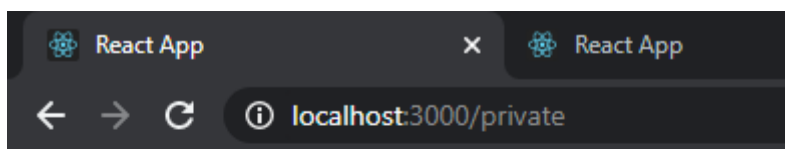


You are not logged in

- [Public Page](#)
- [Private Page](#)

You must log in to view the page at /private

Log in



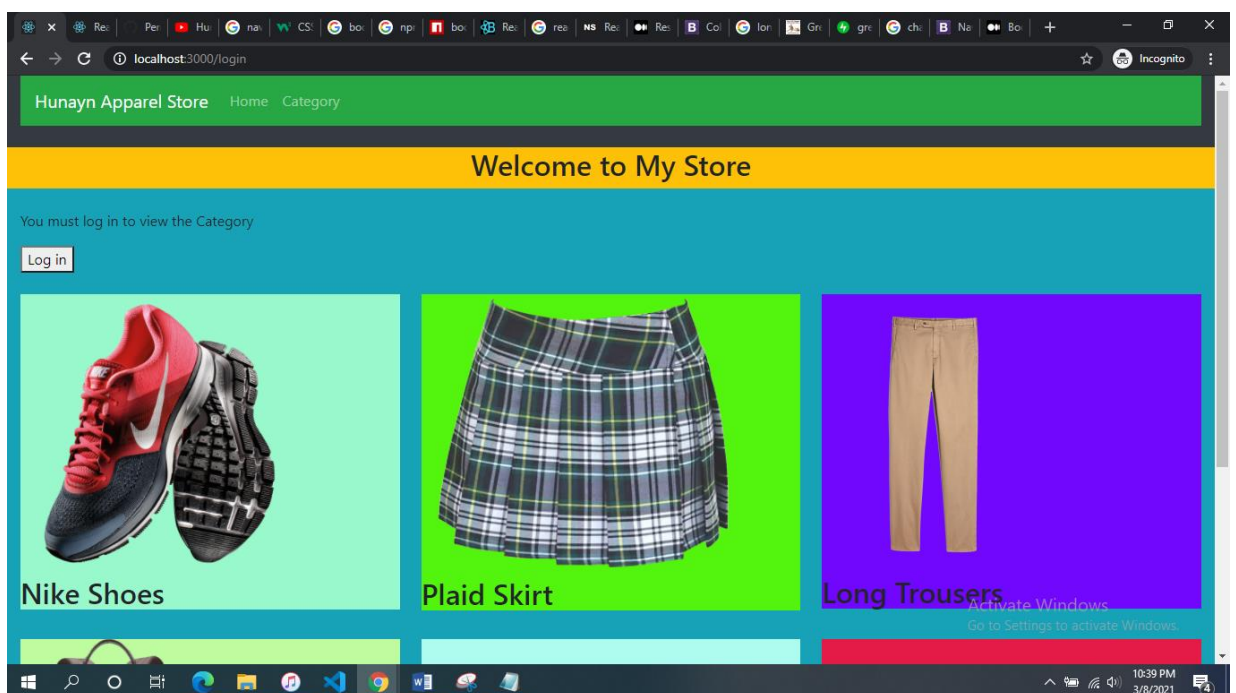
- [Public Page](#)
- [Private Page](#)

Welcome! Sign Out

Private

Soal Latihan Praktikum

Buatlah sebuah situs marketplace(online shop) sederhana, dengan menerapkan konsep SPA (Single Page Application) React Router dimana beberapa menu nya jika di klik terdapat fitur “Nesting” dan “Redirect (Auth)”.



	Link GitHub : https://github.com/hunaynr/framework_prog/tree/main/5thWeek
	Link Youtube : https://www.youtube.com/channel/UCP53BJw16j_nt4Y0sETg1rA

